Computers in Biology and Medicine 188 (2025) 109830

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Addressing imbalance in health data: Synthetic minority oversampling using deep learning

Alex X. Wang [a,1], Viet-Tuan Le [b,1], Hau Nguyen Trung [b], Binh P. Nguyen [a,*]

[a] School of Mathematics and Statistics, Victoria University of Wellington, Kelburn Parade, Wellington 6012, New Zealand
[b] Faculty of Information Technology, Ho Chi Minh City Open University, 97 Vo Van Tan, District 3, Ho Chi Minh City 70000, Viet Nam

ARTICLE INFO

Dataset link: https://github.com/coksvictoria/ACVAE/

Keywords:
Imbalance data
Synthetic data
Deep learning
Contrastive learning

ABSTRACT

Class imbalances in healthcare data, characterized by a disproportionate number of positive cases compared to negative ones, can lead to biased machine learning models that favor the majority class. Ensuring good performance across all classes is crucial for improving healthcare delivery and patient safety. Traditional oversampling methods like SMOTE and its variants face several limitations: they struggle with capturing complex data distributions, handling heterogeneous data types, and natively supporting multi-class datasets. To address these issues, we propose a deep learning based solution using an Auxiliary-guided Conditional Variational Autoencoder (ACVAE) enhanced with contrastive learning. Additionally, we introduce an ensemble technique where ACVAE creates synthetic positive samples, followed by the use of the Edited Centroid-Displacement Nearest Neighbor (ECDNN) algorithm to reduce the majority class. This combined approach takes advantage of ACVAE's ability to produce diverse oversampled data and ECDNN's skill in handling noise through selective undersampling, leading to a more balanced and informative dataset. Our experiments on 12 different health datasets show the effectiveness of our method. We conduct a thorough evaluation of our approach against traditional oversampling techniques and several benchmark machine learning models. The results demonstrate notable improvements in model performance across various metrics, highlighting the potential of deep learning based synthetic oversampling to address class imbalances in healthcare data.

1. Introduction

Class imbalance is a widespread issue in health data, as negative cases frequently exceed positive cases [1]. This imbalance creates significant challenges for machine learning (ML) models, as training on imbalanced data biases the models toward the majority class [2]. Such biases can have serious consequences in healthcare, including misdiagnoses [3], overlooked conditions [4], and ultimately poor patient outcomes [5]. As ML models are increasingly adopted in healthcare to automate processes and reduce labor costs [6,7], it is crucial to ensure that these models perform reliably across all classes to enhance patient safety and improve healthcare delivery [8].

Addressing dataset imbalance is a well-studied problem, with three primary strategies: data-level techniques, algorithm-level techniques, and hybrid approaches [9]. This study focuses on data-level solutions due to their simplicity, computational efficiency, and independence from downstream models [10]. Among data-level techniques, the Synthetic Minority Over-sampling Technique (SMOTE) [11] and its variants are widely employed [12]. These methods balance datasets by generating synthetic samples within the minority class. However, they have notable limitations: they fail to incorporate information from the majority class, which can reduce their effectiveness, and they sometimes introduce noise, failing to capture the complex structures inherent in the data [13]. Additionally, SMOTE and its variants often require workarounds, such as repeated binary training, to handle multi-class data. In contrast, deep generative models (DGMs) are gaining traction for their ability to learn complex data distributions and natively handle multi-class data by generating synthetic samples for all classes simultaneously within a single model. Despite their potential [14], there remains a significant gap in research exploring their application to imbalanced tabular datasets. Existing attempts to apply DGMs to such data have often led to suboptimal performance [15–17], highlighting the need for further investigation in this area.

To address this gap, we propose a new deep learning (DL) based data balancing technique using an Auxiliary-guided Conditional Variational Autoencoder (ACVAE) trained with contrastive learning. Additionally, we investigate an ensemble method where ACVAE generates

* Corresponding author.
E-mail addresses: alex.wang@vuw.ac.nz (A.X. Wang), tuan.lv@ou.edu.vn (V.-T. Le), hau.nt@ou.edu.vn (H.N. Trung), binh.p.nguyen@vuw.ac.nz (B.P. Nguyen).
[1] The authors contribute equally to this work.

https://doi.org/10.1016/j.compbiomed.2025.109830
Received 28 September 2024; Received in revised form 7 January 2025; Accepted 7 February 2025
Available online 20 February 2025

synthetic positive samples, followed by a data undersampling technique. Specifically, we utilize the Edited Centroid-Displacement Nearest Neighbor (ECDNN) algorithm, a method we developed previously [18]. This combined approach takes advantage of ACVAE's ability to produce diverse oversampled data and ECDNN's effectiveness in controlling noise through selective undersampling, resulting in a more balanced and informative dataset. Our comprehensive experiments on 12 diverse health datasets demonstrate the effectiveness of our approach. We conduct an extensive evaluation of our method against traditional oversampling techniques and various benchmark ML classification models. This evaluation provides a detailed analysis of the performance and robustness of our approach. By highlighting the strengths and addressing the limitations of our method, we aim to improve techniques for handling imbalanced datasets in the healthcare field.

The rest of this paper is structured as follows: Section 2 introduces the background and associated work of resampling methods. Section 3 describes the proposed model in detail. Section 4 gives a brief description of datasets and experimental settings. The results and discussion are presented in Section 5, and we draw our conclusions in Section 6.

2. Materials and methods

2.1. Imbalanced data in healthcare

Classification tasks often encounter significant challenges with imbalanced datasets, where rare occurrences or biases in data collection cause ML algorithms to prioritize the majority class, often neglecting minority samples as noise. This imbalance not only results in biased model performance but also imposes substantial costs in healthcare, where the risk of critical misdiagnoses highlights the urgency of addressing underlying data biases. Therefore, ensuring that ML models are not only accurate but also fair is paramount in healthcare applications [19]. Various techniques have been proposed to tackle these challenges through methods such as data-level adjustments (e.g., oversampling, undersampling) and algorithm-level modifications. These approaches emphasize the crucial necessity of rebalancing class distributions to improve the performance of ML models on imbalanced data. In this study, we focus on data-level techniques due to their widespread adoption and effectiveness [20].

2.2. Data balancing techniques

The main approaches for rebalancing imbalanced datasets include oversampling, undersampling, and hybrid sampling [21]. Oversampling involves duplicating or synthesizing samples from the minority class to balance the dataset. Algorithms like Random Oversampling (ROS) and SMOTE create synthetic samples within the minority class, potentially introducing noise. SMOTE variants such as Borderline-SMOTE, SVM-SMOTE, and ADASYN aim to overcome ROS limitations by creating synthetic samples within the area of the minority class [22,23]. Undersampling methods, such as Random Undersampling (RUS), balance the dataset by removing samples from the majority class but may discard valuable information. Informative undersampling techniques like Tomek Links and Edited Nearest Neighbor (ENN) selectively remove instances near the decision boundary to enhance class separation [24]. Both of these techniques identify noisy samples through majority voting of nearest neighbors without considering distance or class information. Our approach, ECDNN, differs by employing centroid displacement for class prediction, which makes it more robust to noisy data. Consequently, ECDNN effectively removes noise and borderline samples, resulting in a smoother decision boundary between classes [18]. Hybrid algorithms, such as SMOTE-Tomek and SMOTE-ENN [25], combine oversampling and undersampling to leverage their respective strengths and mitigate individual drawbacks. These hybrid methods aim to improve classification performance by balancing dataset enrichment and noise minimization, addressing noise propagation issues associated with SMOTE, and enhancing overall classification accuracy [20].

2.3. Conditional variational autoencoder

A Variational Autoencoder (VAE) consists of an encoder network $q_\phi(z|x)$ and a decoder network $p_\theta(x|z)$. The encoder approximates the posterior distribution of latent variables z given input data x, while the decoder models the conditional distribution of data given latent variables. The VAE objective is to maximize the Evidence Lower Bound (ELBO) on the log-likelihood of the observed data:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}[q_\phi(z|x) \parallel p(z)]. \tag{1}$$

The VAE loss function, expressed as the negative ELBO, balances reconstruction accuracy ($\mathbb{E}_{q(z|x)}[\log p(x|z)]$) and latent space regularization ($\mathrm{KL}[q(z|x) \parallel p(z)]$). The reconstruction loss ensures that the model generates outputs similar to the input, while the Kullback–Leibler (KL) divergence term penalizes deviations between the variational posterior $q(z|x)$ and the prior $p(z)$, promoting a structured latent space. Despite its effectiveness, one of the main challenges in VAE optimization is posterior collapse [26], where the learned variational distribution closely resembles the prior. This limits the generative model's capacity because the decoder network cannot effectively utilize all latent dimensions' information [27].

To address posterior collapse, various techniques have been developed to enhance VAE training. Adversarial VAEs employ adversarial training to improve sample quality [28]. Semi-supervised VAEs utilize both labeled and unlabeled data to improve generalization. Hierarchical VAEs use multiple layers of latent variables to capture hierarchical data structures [29]. InfoVAEs maximize mutual information between latent variables and data to learn more informative representations [30]. VampPrior VAEs replace standard Gaussian priors with variational mixture priors for more flexible latent representations [31]. These strategies aim to optimize VAE training by improving sample quality, learning structured representations suitable for specific data characteristics, and enhancing flexibility and information content in the latent space.

2.4. Contrastive learning in VAEs

To enable conditional generation, a crucial requirement is the model's ability to learn latent representations from real data based on labels, a concept referred to as disentanglement. In VAEs, disentanglement involves the model's capability to autonomously identify and separate meaningful factors of variation within datasets [32,33]. Contrastive learning, can be demonstrated by methods such as those discussed by Xie et al. [34], is a self-supervised technique that enhances disentanglement. It achieves this by guiding the model to learn latent representations that are both discriminative (for tasks like contrastive learning) and disentangled (for distinguishing between different generative factors). Training with contrastive objectives encourages models to implicitly promote disentanglement, aiding in the separation of factors within the latent space. Conversely, disentangled representations facilitate contrastive learning by providing a more structured and interpretable latent space where similarities and differences between samples are more significant and meaningful [35]. Furthermore, incorporating labels can further promote a disentangled latent space through auxiliary classification objectives, which optimize subspaces to selectively include or exclude specific label-related information [32]. Mathematically, contrastive learning employs a contrastive loss function, $\mathcal{L}_{\mathrm{Con}}(z_i, z_j, z_k)$, defined as:

$$-\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\exp(\mathrm{sim}(z_i, z_j)/\tau) + \sum_{\substack{k=1 \\ k\neq i}}^{N} \exp(\mathrm{sim}(z_i, z_k)/\tau)}, \tag{2}$$

where z_i and z_j are latent representations for data points x_i and x_j $\mathrm{sim}(z_i, z_j)$ is the similarity measure between them. τ is the temperature parameter controlling the scale of similarity. y is the label, which indicates whether x_i and x_j are from the same class (positive pair) or different classes (negative pair).

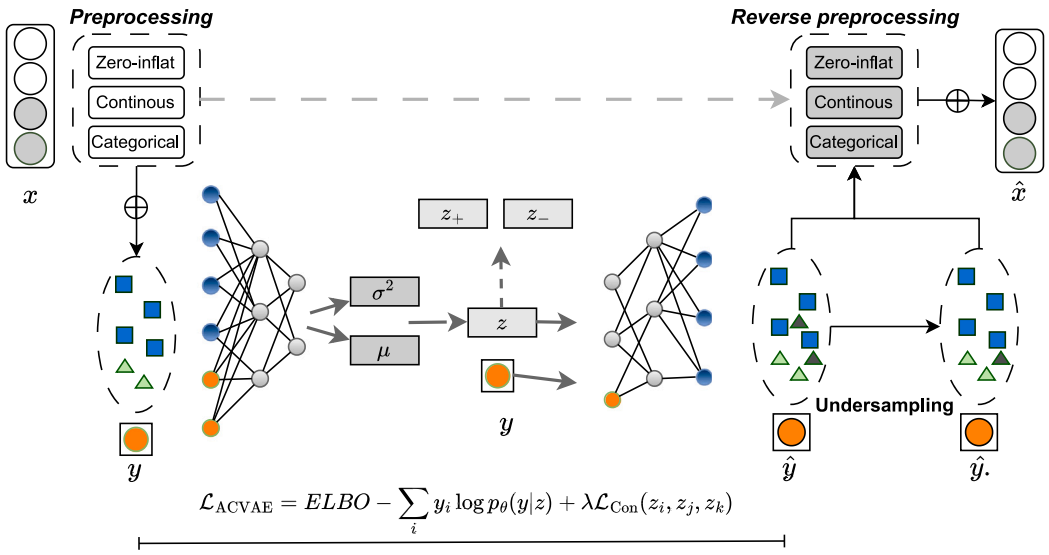

Fig. 1. Auxiliary-guided Conditional Variational Autoencoder (ACVAE). During training, each data sample x undergoes preprocessing and is encoded along with label y to produce a latent space vector z. This vector, along with the label y, serves as input for the decoder, which engages in a reconstruction process incorporating the standard VAE loss, augmented by auxiliary and contrastive losses. These additional losses enhance the model's capacity to learn the latent space in relation to the label. During the generation phase, a randomly sampled latent space vector $\hat{z}$ and label y are inputted into the trained decoder to generate raw synthetic data, which undergoes further refinement through an informative undersampling algorithm before being subjected to reverse processing to restore the data to its original format, $\hat{x}$.

Algorithm 1 Auxiliary Conditional Variational Autoencoder with Contrastive Learning

Input: Real data samples $\{(x(i), y(i))\}_{i=1}^{m}$, learning rates $\alpha_{\text{enc}}, \alpha_{\text{dec}}$, contrastive loss coefficient λ
Initialize: Autoencoder parameters ϕ, θ
for each training iteration **do**
  Sample $(x(i), y(i))$ from the training set
  Sample $z'(i)$ from the true prior p_z
  Sample z from $q_\phi(z|x, y)$
  Train the encoder/decoder (ϕ, θ)**:**
  Compute ELBO with auxiliary classification loss:
  $L_{\text{VAE}} \leftarrow ELBO - \sum_i y_i \log p_\theta(y|z)$
  Compute the contrastive loss based on y:
    $positive\_pairs \leftarrow \|z_i - z_j\|_2$ for $y_i = y_j$
    $negative\_pairs \leftarrow \|z_i - z_k\|_2$ for $y_i = y_k$
  $L_{\text{Con}} \leftarrow \mathcal{L}_{\text{Con}}(z_i, z_j, z_k)$
  Update encoder and decoder parameters:
    $\phi \leftarrow \phi - \alpha_{\text{enc}} \nabla_\phi (L_{\text{VAE}} + \lambda \cdot L_{\text{Con}})$
    $\theta \leftarrow \theta - \alpha_{\text{dec}} \nabla_\theta (L_{\text{VAE}} + \lambda \cdot L_{\text{Con}})$
end for

3. Proposed algorithm

This section introduces ACVAECDNN, our model designed for balancing tabular data, which combines a DL based oversampling technique, ACVAE trained with contrastive learning, and a distance based undersampling technique, ECDNN. As depicted in Fig. 1 and detailed in Algorithm 1, the process consists of two main stages: (1) fine-tuning ACVAE to capture label-dependent feature interactions in the latent space, and (2) utilizing the trained decoder to generate synthetic data, followed by ECDNN to filter out unrealistic samples before reverse processing. We proceed by providing detailed explanations of each component and summarizing our approach.

3.1. Problem definition

The problem of data balancing in a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ as samples from a true data distribution $q(x|y)$ arises when there is a significant class imbalance between the minority $(y_i = 1)$ and the majority $(y_i = 0)$ classes. The goal of a DL based oversampling model is to build neural networks with parameters θ to describe a conditional distribution $p_\theta(x|y)$, ensuring $p_\theta(x|y)$ matches $q(x|y)$ optimally during training. Using $p_\theta(x|y)$, one can generate more minority class samples to balance N_- (negative samples) and N_+ (positive samples) in a binary classification scenario.

3.2. Reversible data preprocessing pipeline

Building upon prior work [36], our preprocessing pipeline addresses the handling of numerical, categorical, and zero-inflated columns commonly found in health events [37]. To process zero-inflated continuous variables, we introduce a binary flag, denoted as γ_j, which indicates whether the value is zero (0) or non-zero (1). This allows us to distinguish between zero and non-zero observations for each row j. To model the continuous values, we apply a variational Gaussian mixture model (VGM) to account for non-Gaussian and multimodal distributions. In this approach, the continuous values are normalized within each mode. Specifically, for the ith mode, the normalized value is represented as $\alpha_{i,j}$, and the corresponding one-hot encoding of the mode is represented as $\beta_{i,j}$, where $\beta_{i,j}$ is a vector that indicates which mode the value belongs to. Thus, for a given row j, the continuous variables are expressed as: $\alpha_{1,j} \oplus \beta_{1,j} \oplus \cdots \oplus \alpha_{n_c,j} \oplus \beta_{n_c,j} \oplus \gamma_j$, where $\alpha_{i,j}$ denotes the normalized value for each mode, $\beta_{i,j}$ represents the one-hot vector encoding the mode, and γ_j is the binary flag for zero values. Following this, categorical features are processed using one-hot encoding, where each unique categorical value is transformed into a binary vector. Discrete columns $D_1, \ldots, D_{n_d}$ are converted into one-hot vectors $d_1, \ldots, d_{n_d}$, with the ith one-hot vector for the jth row denoted as $d_{i,j} = [d_i^{(k)}]$ for $k = 1, \ldots, C_i$, where C_i is the number of categories for feature D_i. Finally, the processed numerical, categorical, and zero-inflated continuous variables are concatenated for each row, which is then ready for use in subsequent DGMs. The representation of the jth row is effectively captured by:

$$r_j = \alpha_{1,j} \oplus \beta_{1,j} \oplus \cdots \oplus \alpha_{n_c,j} \oplus \beta_{n_c,j} \oplus \gamma_j + d_{1,j} \oplus \cdots \oplus d_{n_d,j}.$$

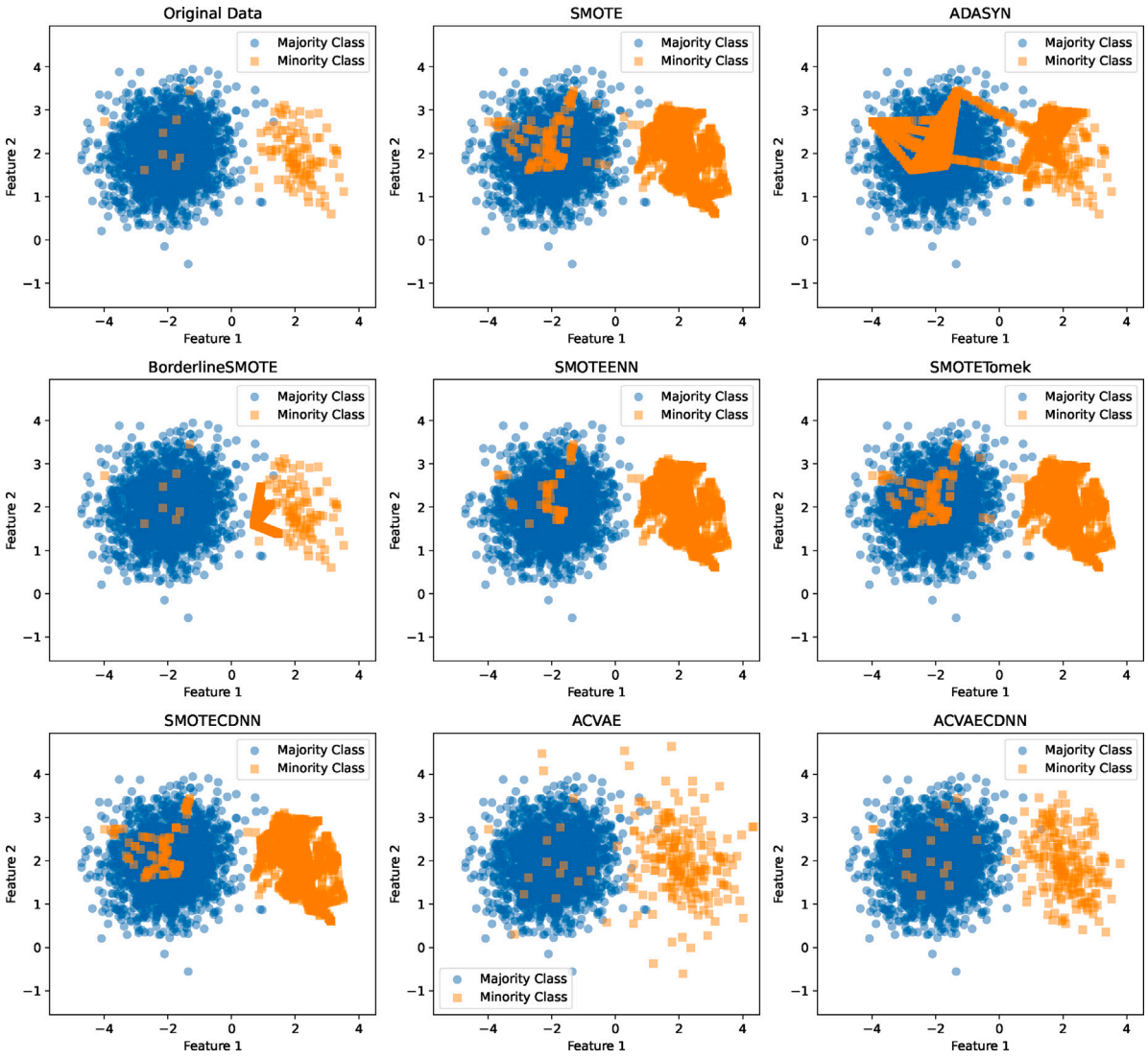

Fig. 2. A visual representation of ACVAECDNN, along with a comparison against other resampling algorithms, using a simulated imbalanced dataset.

3.3. Enhanced loss function

ACVAE is a conditional variational autoencoder designed specifically for tabular data, utilizing class labels to generate diverse synthetic minority samples. It builds upon the standard VAEs, which compresses data into a simplified latent representation and reconstructs it, by incorporating class labels to generate group-specific samples and using contrastive learning to ensure meaningful patterns in the latent space. Contrastive learning encourages the model to group similar data points close together while separating dissimilar ones, improving both the quality and diversity of the generated samples. The loss function of ACVAE incorporates three primary components: ELBO, an auxiliary classification loss term, and a contrastive loss term. The ELBO term includes a reconstruction loss that measures the dissimilarity between input data and its reconstruction given the latent variable and conditioning information. Additionally, the KL divergence regularization ensures that the distribution of latent variables conditioned on the context aligns with a predefined prior distribution. The auxiliary classification loss and contrastive loss encourage similar samples to have latent representations close to each other while pushing dissimilar samples apart in the latent space. Mathematically, the loss function of ACVAE is formulated as:

$$\mathcal{L}_{\text{ACVAE}} = \text{ELBO} - \sum_i y_i \log p_\theta(y|z) + \lambda \mathcal{L}_{\text{Con}}(z_i, z_j, z_k),$$

where ELBO includes reconstruction loss and regularization to ensure accurate data reconstruction and a well-structured latent space. The term y_i represents the true class label, and $p_\theta(y|z)$ is the predicted conditional probability of the class given the latent variable z, with the second term encouraging the model to generate samples matching the correct class. The contrastive loss, $\mathcal{L}_{\text{Con}}(z_i, z_j, z_k)$, promotes meaningful relationships in the latent space by clustering similar samples and separating dissimilar ones, with λ serving as a hyperparameter to control the influence of this term. By integrating these components, ACVAE generates realistic, diverse synthetic data, effectively addressing dataset imbalances and strengthening the performance of ML models.

3.4. ACVAE-ECDNN ensemble

To validate our hypothesis on enhancing data balancing using ACVAE, we created a synthetic binary dataset with an imbalanced class distribution. In Fig. 2, we illustrate the differences between the original dataset and the dataset after applying our oversampling techniques. Traditional oversampling methods often introduce significant noise into the majority class area. These methods typically generate synthetic data that clusters narrowly within the existing minority class region, thereby failing to adequately expand the minority class distribution. This clustering tendency validates the issues discussed earlier, such as limited diversity and increased noise. In contrast, our proposed

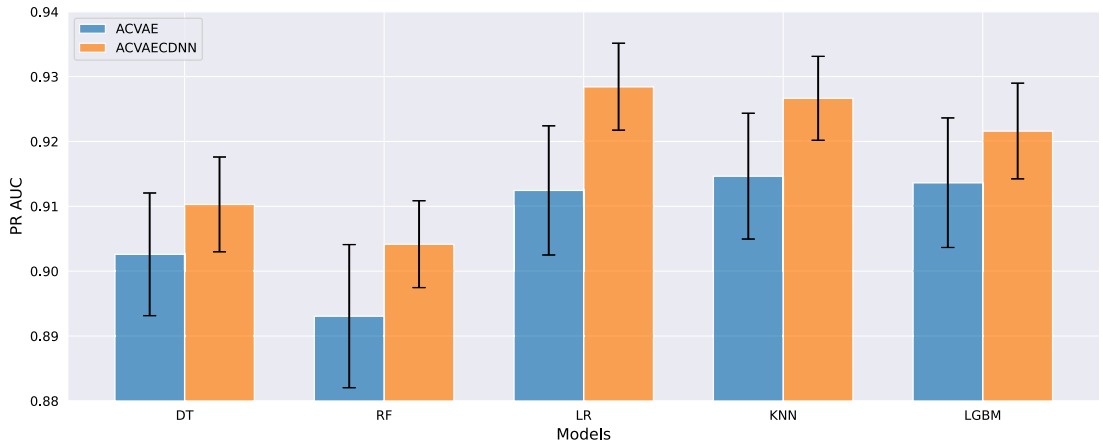

Fig. 3. Comparison of average PR AUC values for ACVAE and ACVAECDNN over 100 Trials on a simulated imbalanced dataset.

ACVAE, trained on both majority and minority classes, demonstrates an improved ability to balance the dataset. ACVAE effectively reduces noise in the majority class area while introducing more diverse synthetic samples into the minority class region. However, it is observed that ACVAE has limitations, such as generating synthetic minority samples that overlap with the majority class area, which can blur class boundaries. Additionally, it may produce samples that are too far away from the actual class boundary, contributing no meaningful value to the dataset. To address these limitations, an informative undersampling technique like ECDNN is needed. Integrating ACVAE with ECDNN can yield clearer boundaries and remove redundant synthetic data, ensuring that the synthetic samples generated by ACVAE are both relevant and useful. This integration is expected to enhance the overall effectiveness of the data balancing process and lead to more accurate and robust models.

To demonstrate our hypothesis on the effectiveness of using informative undersampling for building ensemble models, we conducted a controlled experiment with a synthetic, severely imbalanced binary dataset exhibiting an imbalance ratio of 9:1. We compared the performance of ACVAE and ACVAE with ECDNN across five widely used classification algorithms to highlight the necessity of applying a noise-filtering technique. The results, presented in Fig. 3 as bar charts with variance bars, are based on 100 repeated experimental trials using different randomly generated synthetic data under the same settings. These results clearly show that ACVAECDNN consistently outperforms ACVAE in all cases. Moreover, the performance of ACVAECDNN is not only superior but also more stable, with smaller variance compared to ACVAE. This pattern of superior performance is consistent across all five classification algorithms tested. Therefore, compared to using ACVAE alone, an ensemble model of ACVAE and ECDNN proves to be the optimal choice for enhancing data balancing and achieving better classification performance.

4. Experiments

4.1. Datasets

To thoroughly evaluate our proposed algorithm, we used 12 real-world public datasets from the UCI data repository [38], covering a range of sizes, characteristics, features, and distributions relevant to the health domain. Core details are summarized in Table 1, while more comprehensive information is provided in Appendix A. These datasets vary significantly in size, with the number of rows ranging from 839 to 253,680, and feature both categorical (0 to 94 columns) and numerical data (1 to 32 columns). Imbalance ratios (IR) also vary, indicating different levels of class imbalance, from relatively balanced datasets (IR of 1.09) to significantly imbalanced ones (IR of 10.92). These datasets have been extensively studied in prior ML research [39], ensuring strong benchmarks for comparison across different conditions. This diversity in dataset characteristics ensures a comprehensive evaluation of the algorithm's performance across various real-world scenarios in the health domain.

4.2. Baselines

To validate the effectiveness of ACVAE and ACVAECDNN, we compare them with established benchmark data balancing algorithms: SMOTE, ADASYN, and BorderlineSMOTE from the oversampling family, and SMOTEENN, SMOTETomek, and SMOTECDNN from the hybrid sampling family. In our ablation test, we assess ACVAE and ACVAECDNN alongside their building components: unconditional Tabular VAE (TVAE) and its direct modified version, conditional TVAE (CTVAE). This comparison aims to demonstrate the efficacy of incorporating contrastive learning, auxiliary training, and informative undersampling techniques. Below is a brief overview of each algorithm:

- **SMOTE** [11]**:** is a popular oversampling technique that generates synthetic samples by interpolating between real minority class samples.
- **ADASYN** [23]**:** adaptively generates synthetic samples based on the density distribution of minority class samples, focusing more on regions where the class imbalance is more severe.
- **BoarderlineSMOTE** [22]**:** generates synthetic samples near the decision boundary between minority and majority classes. It focuses on "borderline" samples to improve classification accuracy for difficult cases.
- **SMOTEENN** [25]**:** combines the oversampling strategy of SMOTE with the undersampling strategy of Edited Nearest Neighbors (ENN) to achieve a better balance between minority and majority class samples.
- **SMOTETomek** [25]**:** combines SMOTE with Tomek Links, generating synthetic samples and then removing borderline cases that are hard to classify. This approach cleans the dataset by eliminating ambiguous samples.
- **SMOTECDNN** [18]**:** is a hybrid method that combines SMOTE with Edited Centroid-Displacement Nearest Neighbor (ECDNN). It generates synthetic minority class samples using SMOTE and then applies ECDNN to filter and remove noisy samples. Our previous study [18] demonstrated that SMOTECDNN outperforms other hybrid methods like SMOTEENN and SMOTETomek, offering a more robust approach to handling class imbalances by enhancing data quality and reducing noise.

Table 1
Details of the datasets used in this study. #Rows, #Num and #Cat represents the number of samples, numerical columns and categorical columns. IR stands for Imbalance Ratio.

Dataset ID	Dataset name	#Rows	#Cat	#Num	IR
759	Glioma Grading Clinical and Mutation Features	839	22	1	1.38
863	Maternal Health Risk	1014	0	6	1.49
503	Hepatitis C Virus (HCV) for Egyptian patients	1385	9	19	1.09
030	Contraceptive Method Choice	1473	7	2	1.89
579	Myocardial Infarction Complications	1700	94	17	9.00
193	Cardiotocography	2126	0	21	10.92
890	AIDS Clinical Trials Group Study 175	2139	11	12	3.11
887	National Health and Nutrition Health Survey 20	2278	0	7	5.26
880	SUPPORT2	9105	10	32	2.14
264	EEG Eye State	14 980	0	14	1.23
296	Diabetes 130-US Hospitals for Years 1999–2008	101 766	39	8	4.83
891	CDC Diabetes Health Indicators	253 680	14	7	6.18

- **TVAE** [36]: uses a Variational Autoencoder (VAE) to generate synthetic tabular data. Unlike ACVAE, TVAE lacks a built-in conditional generation function. To adapt TVAE for oversampling tasks, we implemented a method where we iterate through the generated synthetic data and selectively retain only positive samples in each iteration. This process is repeated until we accumulate enough positive samples to achieve a balanced dataset. Additionally, we developed Conditional TVAE (**CTVAE**), a direct modification of TVAE that incorporates additional conditioning information for data balancing. The training process and loss function in CTVAE remain the same as in TVAE.

4.3. Evaluation settings

To thoroughly evaluate the robustness of the proposed algorithms, we utilized five well-known classification techniques: Decision Tree (DT), *k*-Nearest Neighbors (*k*-NN), Random Forest (RF), Logistic Regression (LR), and Light Gradient Boosting Machine (LightGBM) [40]. These methods were chosen to represent a diverse spectrum of approaches, from straightforward linear models to sophisticated ensemble learning techniques. A more detailed overview of these algorithms can be found in [41]. Moreover, these classifiers exhibit different levels of sensitivity to class imbalances. Thus, our objective was to thoroughly evaluate the robustness of the proposed algorithms across diverse scenarios, demonstrating their efficacy in enhancing classification performance while remaining resilient to different classification algorithms. To ensure a fair comparison, we maintained consistent default hyperparameters for each classification algorithm. To prevent data leakage, we split each dataset into 80% for training and 20% for testing. All models were trained under identical conditions with default parameters to maintain consistency. To ensure reliability, each experiment was repeated five times using different random seeds for data splitting, and average performance metrics were reported to assess model effectiveness [42]. Given that all datasets have varying levels of imbalance, we used the Area Under the Precision–Recall Curve (PRAUC) as our primary metric and also reported the average Area Under the Receiver Operating Characteristic Curve (AUCROC), Balanced Accuracy (BA), and F1 Score [43]. This approach guarantees the credibility and applicability of our findings in evaluating the proposed algorithm against established methods. All the experiments were conducted on a server equipped with an Intel Xeon CPU at 2.30 GHz, 32 GB of RAM, and an NVIDIA T4 2560 GPU with 16 GB of memory.

5. Results and discussion

5.1. Comparative study on different datasets

The average PRAUC and ROCAUC scores across 12 datasets are presented in Table 2, comparing eight data balancing algorithms alongside the original imbalanced datasets. Each dataset is categorized by size, listed in increasing order, and assigned a unique identifier. In addition to the PRAUC and ROCAUC scores, where higher values indicate better performance, rankings are provided in parentheses, with lower numbers signifying superior performance.

As shown in Table 2, ACVAE and ACVAECDNN consistently achieve competitive PRAUC scores across most datasets, with ACVAECDNN outperforming other methods on average (0.594 with an average ranking of 2.4). Upon further analysis, ACVAECDNN achieves the best results in 8 out of 12 datasets, particularly excelling in larger datasets like D891, where it significantly outperforms alternative methods. In datasets such as D030, D579, and D296, where other algorithms fail to improve upon the original data's performance, ACVAECDNN is the only algorithm that provides an enhancement. These results highlight the effectiveness of ACVAECDNN in generating synthetic minority instances that improve model performance, demonstrating the efficiency of the ensemble approach compared to traditional data balancing methods. However, it is worth noting that ACVAE and ACVAECDNN perform poorly on datasets D863, D193, and D264, which consist solely of numerical variables. This suggests that these datasets may be less complex and may be better suited for traditional models, as the powerful design of ACVAE for handling mixed data types is not fully utilized here.

A similar trend is observed across other evaluation metrics, including ROCAUC, BA, and F1 score. ACVAECDNN achieves the highest average ROCAUC (0.771 with an average ranking of 2.7), demonstrating its strong ability to discriminate between classes, particularly in datasets with mixed data types. In terms of BA, ACVAECDNN also performs impressively, with an average score of 0.686 and an overall ranking of 2.6. This metric highlights ACVAECDNN's ability to maintain a balanced representation of both minority and majority classes, especially in situations involving class imbalances. Similarly, ACVAECDNN achieves an average F1 score of 0.603, ranking 2.9, reflecting its strong performance in both precision and recall of the minority class. These results illustrate that ACVAECDNN's ensemble approach not only enhances overall classification performance but also ensures balanced representation of both classes across multiple evaluation metrics. In comparison, other methods tend to excel in only one or two areas, while ACVAECDNN offers consistent improvements across all metrics, making it a comprehensive and reliable data balancing method.

5.2. Comparative study on different classifiers

To explore the impact of data balancing techniques on various classification algorithms, we present the average PRAUC, ROCAUC, BA, and F1 scores for five classifiers in Tables 3. These tables compare the performance of five widely used classification algorithms on both original and balanced datasets. Each classifier exhibits varying levels of complexity and sensitivity to imbalanced classes. Average scores and rankings are provided to offer a comprehensive understanding of their comparative performance.

As shown in Table 3, ACVAECDNN emerges as the top performer on average, achieving a PRAUC of 0.594 with an average ranking of

Table 2
The values of ML utility computed w.r.t. PRAUC, ROCAUC, Balanced accuracy & F1 across 12 datasets, where higher means better.

Dataset	Origin	SMOTE	ADASYN	BorderlineSMOTE	SMOTEENN	SMOTETomek	SMOTECDNN	ACVAE	ACVAECDNN
PRAUC									
D759	0.810 (7)	0.806 (9)	0.810 (8)	0.817 (6)	0.825 (4)	0.824 (5)	0.826 (3)	0.829 (2)	0.833 (1)
D863	0.830 (4)	0.835 (3)	0.842 (1)	0.830 (5)	0.819 (9)	0.820 (8)	0.838 (2)	0.822 (7)	0.828 (6)
D503	0.228 (9)	0.232 (8)	0.235 (7)	0.246 (4)	0.253 (2)	0.235 (6)	0.243 (5)	0.251 (3)	0.262 (1)
D030	0.658 (2)	0.651 (5)	0.654 (3)	0.654 (4)	0.619 (9)	0.633 (8)	0.645 (6)	0.638 (7)	0.659 (1)
D579	0.202 (2)	0.192 (7)	0.198 (5)	0.190 (9)	0.200 (3)	0.192 (7)	0.195 (6)	0.199 (4)	0.217 (1)
D193	0.809 (1)	0.792 (3)	0.794 (2)	0.781 (6)	0.751 (9)	0.791 (4)	0.754 (7)	0.754 (8)	0.782 (5)
D890	0.723 (4)	0.724 (3)	0.708 (8)	0.723 (5)	0.691 (9)	0.726 (2)	0.718 (6)	0.713 (7)	0.733 (1)
D887	0.276 (9)	0.281 (8)	0.282 (7)	0.300 (2)	0.301 (1)	0.288 (6)	0.288 (5)	0.288 (4)	0.294 (3)
D880	0.878 (5)	0.848 (9)	0.875 (7)	0.875 (6)	0.882 (3)	0.875 (8)	0.882 (4)	0.888 (2)	0.898 (1)
D264	0.835 (1)	0.831 (6)	0.833 (4)	0.834 (2)	0.804 (9)	0.832 (5)	0.833 (3)	0.815 (8)	0.820 (7)
D296	0.437 (2)	0.431 (9)	0.435 (5)	0.435 (6)	0.431 (8)	0.433 (7)	0.435 (4)	0.437 (3)	0.448 (1)
D891	0.337 (5)	0.329 (6)	0.326 (9)	0.329 (7)	0.348 (2)	0.328 (8)	0.339 (4)	0.339 (3)	0.358 (1)
Average	0.585 (4.2)	0.579 (6.3)	0.583 (5.5)	0.585 (5.1)	0.577 (5.6)	0.581 (6.1)	0.583 (4.5)	0.581 (4.8)	0.594 (2.4)
ROCAUC									
D759	0.959 (2)	0.949 (5)	0.953 (3)	0.944 (6)	0.944 (7)	0.949 (4)	0.938 (9)	0.943 (8)	0.960 (1)
D863	0.876 (2)	0.875 (5)	0.876 (3)	0.877 (1)	0.855 (7)	0.875 (4)	0.864 (6)	0.833 (9)	0.836 (8)
D503	0.575 (8)	0.570 (9)	0.582 (5)	0.579 (7)	0.581 (6)	0.589 (4)	0.608 (3)	0.613 (2)	0.640 (1)
D030	0.697 (2)	0.694 (4)	0.694 (3)	0.686 (5)	0.678 (8)	0.677 (9)	0.678 (7)	0.680 (6)	0.703 (1)
D579	0.469 (9)	0.476 (8)	0.484 (6)	0.491 (4)	0.493 (3)	0.482 (7)	0.490 (5)	0.493 (2)	0.515 (1)
D193	0.657 (6)	0.647 (8)	0.660 (3)	0.661 (2)	0.663 (1)	0.647 (8)	0.655 (7)	0.657 (5)	0.659 (4)
D890	0.891 (9)	0.895 (6)	0.894 (7)	0.893 (8)	0.914 (2)	0.903 (5)	0.909 (4)	0.914 (3)	0.917 (1)
D887	0.903 (4)	0.905 (2)	0.911 (1)	0.904 (3)	0.813 (9)	0.892 (5)	0.856 (7)	0.855 (8)	0.856 (6)
D880	0.799 (2)	0.793 (6)	0.791 (7)	0.790 (9)	0.798 (3)	0.791 (8)	0.794 (5)	0.798 (3)	0.806 (1)
D264	0.667 (9)	0.678 (8)	0.683 (7)	0.688 (2)	0.692 (1)	0.687 (3)	0.684 (6)	0.686 (4)	0.684 (5)
D296	0.876 (9)	0.885 (6)	0.878 (8)	0.889 (3)	0.891 (2)	0.881 (7)	0.887 (5)	0.887 (4)	0.898 (1)
D891	0.756 (5)	0.755 (7)	0.754 (9)	0.755 (6)	0.777 (2)	0.754 (8)	0.762 (4)	0.771 (3)	0.780 (1)
Average	0.760 (5.6)	0.760 (6.0)	0.763 (4.8)	0.763 (4.5)	0.758 (4.4)	0.761 (5.8)	0.760 (5.8)	0.761 (4.9)	0.771 (2.7)
Balanced accuracy									
D759	0.887 (1)	0.870 (5)	0.879 (2)	0.870 (6)	0.849 (9)	0.875 (3)	0.851 (8)	0.854 (7)	0.874 (4)
D863	0.861 (2)	0.855 (5)	0.862 (1)	0.859 (3)	0.837 (7)	0.856 (4)	0.854 (6)	0.825 (9)	0.833 (8)
D503	0.508 (8)	0.503 (9)	0.511 (5)	0.510 (6)	0.509 (7)	0.513 (4)	0.522 (3)	0.528 (2)	0.545 (1)
D030	0.679 (3)	0.675 (4)	0.680 (2)	0.673 (5)	0.653 (9)	0.656 (8)	0.663 (7)	0.665 (6)	0.682 (1)
D579	0.349 (9)	0.356 (8)	0.361 (7)	0.369 (4)	0.376 (2)	0.361 (6)	0.367 (5)	0.373 (3)	0.392 (1)
D193	0.430 (5)	0.420 (9)	0.432 (2)	0.427 (7)	0.432 (3)	0.420 (8)	0.429 (6)	0.431 (4)	0.439 (1)
D890	0.853 (9)	0.855 (7)	0.854 (8)	0.860 (6)	0.873 (3)	0.872 (4)	0.868 (5)	0.874 (2)	0.879 (1)
D887	0.870 (4)	0.879 (2)	0.883 (1)	0.874 (3)	0.820 (9)	0.859 (5)	0.855 (6)	0.842 (8)	0.850 (7)
D880	0.844 (4)	0.826 (9)	0.838 (7)	0.838 (6)	0.847 (3)	0.836 (8)	0.844 (5)	0.850 (2)	0.853 (1)
D264	0.475 (9)	0.480 (8)	0.483 (7)	0.497 (1)	0.497 (2)	0.491 (4)	0.488 (6)	0.488 (5)	0.493 (3)
D296	0.803 (7)	0.807 (3)	0.797 (8)	0.807 (2)	0.794 (9)	0.806 (5)	0.806 (4)	0.805 (6)	0.819 (1)
D891	0.550 (5)	0.546 (6)	0.542 (8)	0.545 (7)	0.564 (2)	0.541 (9)	0.554 (4)	0.557 (3)	0.570 (1)
Average	0.676 (5.5)	0.673 (6.2)	0.677 (4.5)	0.677 (4.4)	0.671 (5.7)	0.674 (5.3)	0.675 (5.5)	0.674 (4.9)	0.686 (2.6)
F1									
D759	0.820 (1)	0.805 (2)	0.800 (4)	0.798 (5)	0.765 (7)	0.803 (3)	0.756 (9)	0.760 (8)	0.790 (6)
D863	0.843 (4)	0.832 (7)	0.849 (1)	0.840 (5)	0.812 (9)	0.847 (2)	0.847 (3)	0.826 (8)	0.834 (6)
D503	0.447 (2)	0.436 (7)	0.439 (6)	0.444 (3)	0.435 (8)	0.441 (4)	0.435 (9)	0.440 (5)	0.455 (1)
D030	0.665 (3)	0.664 (4)	0.664 (5)	0.670 (1)	0.625 (9)	0.635 (8)	0.657 (6)	0.638 (7)	0.668 (2)
D579	0.231 (9)	0.237 (8)	0.239 (7)	0.247 (4)	0.254 (2)	0.240 (6)	0.245 (5)	0.252 (3)	0.268 (1)
D193	0.203 (4)	0.193 (9)	0.199 (5)	0.193 (8)	0.203 (3)	0.194 (7)	0.197 (6)	0.203 (2)	0.221 (1)
D890	0.830 (5)	0.816 (8)	0.814 (9)	0.827 (7)	0.838 (3)	0.833 (4)	0.830 (6)	0.848 (1)	0.844 (2)
D887	0.849 (4)	0.855 (3)	0.859 (1)	0.842 (5)	0.837 (6)	0.829 (8)	0.856 (2)	0.823 (9)	0.835 (7)
D880	0.898 (3)	0.849 (9)	0.895 (4)	0.890 (8)	0.893 (5)	0.890 (7)	0.892 (6)	0.901 (2)	0.918 (1)
D264	0.278 (9)	0.284 (8)	0.286 (7)	0.308 (1)	0.305 (2)	0.294 (4)	0.290 (6)	0.293 (5)	0.301 (3)
D296	0.733 (3)	0.729 (5)	0.713 (8)	0.723 (6)	0.704 (9)	0.742 (1)	0.720 (7)	0.729 (4)	0.738 (2)
D891	0.345 (3)	0.334 (7)	0.326 (9)	0.330 (8)	0.350 (2)	0.335 (6)	0.342 (5)	0.344 (4)	0.365 (1)
Average	0.595 (4.2)	0.586 (6.3)	0.590 (5.1)	0.593 (4.8)	0.585 (5.7)	0.590 (4.9)	0.589 (5.9)	0.588 (4.9)	0.603 (2.9)

1.4. ACVAECDNN outperformed other methods across all classifiers, except for LGBM, where it ranked third with an average PRAUC of 0.646, which is very close to the top performer, BorderlineSMOTE with 0.649. Notably, ACVAECDNN demonstrates exceptional performance with simpler classification algorithms, particularly LR, where it increases the average PRAUC from 0.481 (original data) to 0.502. This suggests that data-level balancing techniques like ACVAECDNN are more effective with simpler models, while more advanced algorithms like LGBM, which are less sensitive to class imbalance, show smaller improvements. Additionally, ACVAECDNN's lower performance on datasets with only numerical variables may explain its reduced effectiveness with LGBM.

A similar trend is observed across other evaluation metrics. For ROCAUC, ACVAECDNN achieved an average score of 0.771 with an average ranking of 1.2, consistently outperforming all classifiers except for LGBM, where it ranks second. This strong ability to restore clear class distributions is further demonstrated by its performance in BA and F1 score. ACVAECDNN achieved an average BA ranking of 1.4 and an average F1 score ranking of 1.8, both of which reflect its balanced performance in identifying both majority and minority classes. These results demonstrate the robustness of ACVAECDNN across multiple evaluation metrics, highlighting its effectiveness in generating synthetic minority instances that significantly improve model performance compared to traditional data balancing methods.

Table 3
The values of ML utility computed w.r.t. PRAUC, ROCAUC, Balanced accuracy & F1 across 5 classifiers, where higher means better.

Classifier	Origin	SMOTE	ADASYN	BorderlineSMOTE	SMOTEENN	SMOTETomek	SMOTECDNN	ACVAE	ACVAECDNN
PRAUC									
LR	0.481 (9)	0.489 (5)	0.494 (2)	0.491 (3)	0.49 (4)	0.488 (6)	0.484 (7)	0.484 (8)	0.502 (1)
KNN	0.545 (2)	0.535 (8)	0.536 (5)	0.539 (3)	0.536 (7)	0.537 (4)	0.534 (9)	0.536 (6)	0.559 (1)
DT	0.662 (3)	0.644 (9)	0.658 (6)	0.661 (4)	0.655 (7)	0.661 (5)	0.663 (2)	0.654 (8)	0.670 (1)
RF	0.592 (2)	0.586 (5)	0.588 (3)	0.584 (7)	0.576 (9)	0.584 (8)	0.588 (4)	0.585 (6)	0.593 (1)
LGBM	0.645 (4)	0.643 (6)	0.641 (7)	0.649 (1)	0.629 (9)	0.637 (8)	0.648 (2)	0.644 (5)	0.646 (3)
Average	0.585 (4.0)	0.579 (6.6)	0.583 (4.6)	0.585 (3.6)	0.577 (7.2)	0.581 (6.2)	0.583 (4.8)	0.581 (6.6)	0.594 (1.4)
ROCAUC									
LR	0.691 (9)	0.697 (8)	0.699 (4)	0.697 (7)	0.710 (2)	0.698 (6)	0.704 (3)	0.698 (5)	0.714 (1)
KNN	0.743 (3)	0.738 (7)	0.745 (2)	0.740 (6)	0.728 (9)	0.742 (4)	0.735 (8)	0.741 (5)	0.749 (1)
DT	0.807 (4)	0.803 (9)	0.811 (2)	0.806 (5)	0.803 (8)	0.807 (3)	0.805 (6)	0.805 (7)	0.812 (1)
RF	0.759 (8)	0.760 (5)	0.757 (9)	0.761 (3)	0.759 (7)	0.761 (2)	0.760 (4)	0.760 (6)	0.774 (1)
LGBM	0.802 (5)	0.803 (4)	0.805 (3)	0.813 (1)	0.792 (9)	0.796 (8)	0.797 (7)	0.801 (6)	0.807 (2)
Average	0.760 (5.8)	0.760 (6.6)	0.763 (4.0)	0.763 (4.4)	0.758 (7.0)	0.761 (4.6)	0.760 (5.6)	0.761 (5.8)	0.771 (1.2)
Balanced accuracy									
LR	0.581 (9)	0.594 (8)	0.600 (3)	0.595 (6)	0.604 (2)	0.597 (5)	0.599 (4)	0.594 (7)	0.613 (1)
KNN	0.650 (2)	0.641 (7)	0.645 (3)	0.642 (5)	0.635 (9)	0.641 (6)	0.636 (8)	0.644 (4)	0.657 (1)
DT	0.742 (1)	0.729 (9)	0.740 (4)	0.739 (5)	0.730 (8)	0.734 (7)	0.741 (3)	0.735 (6)	0.742 (2)
RF	0.679 (3)	0.675 (7)	0.678 (4)	0.675 (6)	0.669 (9)	0.679 (2)	0.676 (5)	0.673 (8)	0.688 (1)
LGBM	0.726 (4)	0.724 (6)	0.725 (5)	0.734 (1)	0.714 (9)	0.719 (8)	0.728 (3)	0.724 (7)	0.729 (2)
Average	0.676 (3.8)	0.673 (7.4)	0.677 (3.8)	0.677 (4.6)	0.671 (7.4)	0.674 (5.6)	0.675 (4.6)	0.674 (6.4)	0.686 (1.4)
F1									
LR	0.492 (8)	0.497 (5)	0.497 (6)	0.502 (2)	0.501 (3)	0.499 (4)	0.494 (7)	0.488 (9)	0.503 (1)
KNN	0.552 (2)	0.542 (8)	0.541 (9)	0.543 (7)	0.545 (5)	0.548 (3)	0.545 (6)	0.546 (4)	0.573 (1)
DT	0.670 (4)	0.649 (9)	0.668 (6)	0.676 (2)	0.669 (5)	0.672 (3)	0.662 (7)	0.661 (8)	0.686 (1)
RF	0.608 (1)	0.594 (7)	0.594 (6)	0.593 (8)	0.579 (9)	0.595 (5)	0.596 (3)	0.595 (4)	0.602 (2)
LGBM	0.652 (3)	0.645 (6)	0.645 (7)	0.652 (2)	0.629 (9)	0.638 (8)	0.653 (1)	0.649 (5)	0.650 (4)
Average	0.595 (3.6)	0.586 (7)	0.590 (6.8)	0.593 (4.2)	0.585 (6.2)	0.590 (4.6)	0.589 (4.8)	0.588 (6)	0.603 (1.8)

Table 4
Average training and sampling time on the *D891*-CDC Diabetes Health Indicators Dataset.

	SMOTE	ADASYN	BorderlineSMOTE	SMOTEENN	SMOTETomek	SMOTECDNN	ACVAE	ACVAECDNN
Time	16 s	37 s	36 s	2 m 21 s	2 m 11 s	4 m 21 s	1 h 19 m	1 h 19 m + 3 m 22 s

5.3. Ablation study

To understand the efficiency of each component of ACVAECDNN, including the conditional architecture, training guided by contrastive learning and auxiliary loss, and the application of informative undersampling, we present the results of unconditional TVAE, CTVAE, ACVAE, and ACVAECDNN in Fig. 4. In the radar plot, the average PRAUC values are on the *y*-axis, with higher values indicating better performance. The plot shows that simply adding labels to the original TVAE to create CTVAE leads to significantly worse outcomes. This is because, with ELBO only and no additional regulation, the label is not strong enough to guide the TVAE in learning a meaningful latent space based on the label. On the other hand, ACVAE, which uses auxiliary loss and contrastive loss as regulation terms, shows significantly improved performance over CTVAE. This indicates that using appropriate regulation terms can considerably enhance TVAE's conditional generation capabilities. Furthermore, informative undersampling further improves the performance, as demonstrated by ACVAECDNN, which consistently achieves the best PRAUC among these models across all five classification algorithms. This highlights the effectiveness of the ensemble approach in boosting model performance.

5.4. Computational efficiency

In real-world applications, computational cost is a key consideration, especially with large datasets. For smaller datasets, both simple and complex algorithms tend to have similar execution times, but as the dataset size increases, computational costs rise exponentially. To illustrate this, Table 4 presents the average training and sampling times for the *D891* dataset, the largest in this study. As expected, Non DL models like SMOTE and its variants are significantly faster, generating synthetic data during training; for example, SMOTE takes just 16 s, while ADASYN and BorderlineSMOTE require 37 and 36 s, respectively. More complex models, such as SMOTEENN, SMOTETomek, and SMOTECDNN, which are ensemble models, take longer. DL based methods like ACVAE require substantial computational power, taking 1 h and 19 min for training. However, a key advantage of DGMs like ACVAE is that once trained, they can generate an unlimited amount of new data. ACVAECDNN, an ensemble built on the pre-trained ACVAE, requires an additional 3 min and 22 s for filtering and sampling. These results highlight the trade-off between model complexity and computational efficiency, where simpler models offer faster processing times but may not achieve the same level of performance as more complex DL models.

6. Conclusion

Class imbalance is a significant challenge in healthcare data, leading to biased machine learning models that can negatively affect patient outcomes. Traditional data balancing methods, such as SMOTE and its variants, have limitations, including the introduction of noise and an inability to capture complex data structures. Although Deep Generative Models (DGMs) offer a promising solution, applying them to imbalanced tabular datasets has proven difficult and often results in suboptimal performance. To address this challenge, we proposed ACVAE and its enhanced version with informative undersampling,

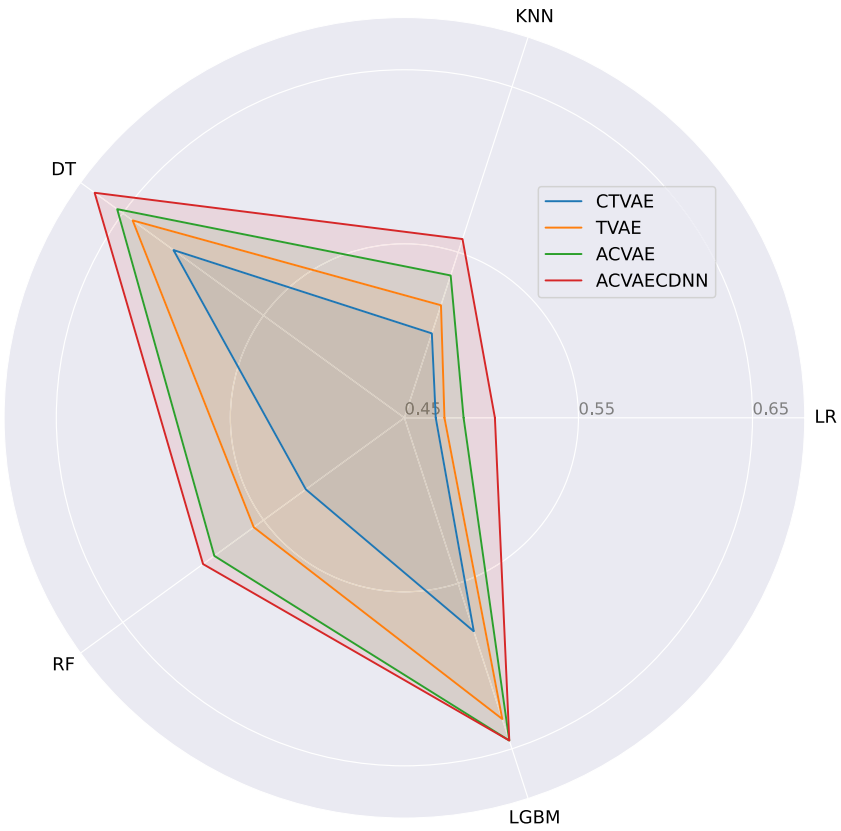

Fig. 4. Competitive analysis of DL based algorithms in terms of average PRAUC, with values lower than 0.45 filtered out.

ACVAECDNN. Through a comprehensive evaluation across 12 diverse health datasets, we demonstrated that ACVAECDNN effectively generates synthetic minority samples, improving model performance and surpassing traditional data balancing techniques.

However, our approach has some limitations. ACVAECDNN tends to perform less effectively with datasets composed solely of numerical variables, and the computational cost associated with training DGMs can be high, particularly when working with large-scale datasets. To address these issues, we plan to refine our algorithm's performance on numerical datasets and develop more computationally efficient undersampling techniques to further optimize performance and reduce computational costs. This will enhance the scalability and applicability of our approach in real-world healthcare settings.

CRediT authorship contribution statement

Alex X. Wang: Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Viet-Tuan Le:** Writing – review & editing, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Hau Nguyen Trung:** Writing – review & editing, Investigation, Formal analysis. **Binh P. Nguyen:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they do not have any competing financial interests or personal relationships that could have influenced the findings presented in this manuscript.

Acknowledgments

This research was supported by Ho Chi Minh City Open University under the grant number E2024.04.1.

Appendix A. Additional dataset information

We utilized 12 datasets from the UCI Machine Learning Repository.[2] Detailed descriptions of the features for each dataset, including their specific attributes beyond general categorizations (e.g., categorical vs. numerical), are provided below.

- Glioma Grading Clinical and Mutation Features (UCI)
- Maternal Health Risk (UCI)
- Hepatitis C Virus (HCV) for Egyptian Patients (UCI)
- Contraceptive Method Choice (UCI)
- Myocardial Infarction Complications (UCI)
- Cardiotocography (UCI)
- AIDS Clinical Trials Group Study 175 (UCI)
- National Health and Nutrition Health Survey 20 (UCI)
- SUPPORT2 (UCI)
- EEG Eye State (UCI)
- Diabetes 130-US Hospitals for Years 1999-2008 (UCI)
- CDC Diabetes Health Indicators (UCI)

Appendix B. Details of experimental implementation

DL models tend to be slower compared to our baseline traditional models. To ensure a fair comparison, we maintained the same set of hyperparameters across different datasets. The detailed architecture of the ACVAE model is described in Section 3 of the main text. Below are the specific hyperparameter settings used:

- Embedding dimension: 256
- Compression dimensions: (128, 128)

[2] https://archive.ics.uci.edu/datasets.

- Decompression dimensions: (128, 128)
- Batch size: 64
- Early stopping threshold: 100
- Number of epochs: 1000

For the classification task, we used the default hyperparameters provided by the scikit-learn package. These default settings can be accessed and reviewed in detail on the official scikit-learn website.[3]

Appendix C. Performance metrics formula

In this study, all evaluation metrics are derived from the confusion matrix, a tool frequently used to assess classifier performance, where true positives (TP) and false positives (FP) are represented. The ROCAUC is generated by plotting sensitivity (SN) against 1-specificity (SP), while the PRAUC plots precision against recall, where

$$SN = \frac{TP}{TP + FN} \qquad SP = \frac{TN}{TN + FP} \tag{C.1}$$

$$Balanced\,Accuracy = \frac{SN + SP}{2} \tag{C.2}$$

$$F1 - score = \frac{2 * TP}{2 * TP + FN + FP} \tag{C.3}$$

Data availability

Data and code are available in the following GitHub repository: https://github.com/coksvictoria/ACVAE/.

References

[1] J.M. Johnson, T.M. Khoshgoftaar, Survey on deep learning with class imbalance, J. Big Data 6 (1) (2019) 1–54.
[2] S. Das, S.S. Mullick, I. Zelinka, On supervised class-imbalanced learning: An updated perspective and some key challenges, IEEE Trans. Artif. Intell. 3 (6) (2022) 973–993.
[3] S. Wang, X. Zhu, Predictive modeling of hospital readmission: challenges and solutions, IEEE/ ACM Trans. Comput. Biol. Bioinform. 19 (5) (2021) 2975–2995.
[4] N. Norori, Q. Hu, F.M. Aellen, F.D. Faraci, A. Tzovara, Addressing bias in big data and AI for health care: A call for open science, Patterns 2 (10) (2021).
[5] I. Araf, A. Idri, I. Chairi, Cost-sensitive learning for imbalanced medical data: a review, Artif. Intell. Rev. 57 (4) (2024) 80.
[6] Z. Abbas, M.U. Rehman, H. Tayara, K.T. Chong, ORI-Explorer: a unified cell-specific tool for origin of replication sites prediction by feature fusion, Bioinformatics 39 (11) (2023) btad664.
[7] Z. Abbas, M.U. Rehman, H. Tayara, S.W. Lee, K.T. Chong, m5C-Seq: Machine learning-enhanced profiling of RNA 5-methylcytosine modifications, Comput. Biol. Med. 182 (2024) 109087.
[8] M. Javaid, A. Haleem, R.P. Singh, R. Suman, S. Rab, Significance of machine learning in healthcare: Features, pillars and applications, Int. J. Intell. Netw. 3 (2022) 58–73.
[9] M. Saini, S. Susan, Tackling class imbalance in computer vision: a contemporary review, Artif. Intell. Rev. 56 (Suppl 1) (2023) 1279–1335.
[10] A.X. Wang, C.R. Simpson, B.P. Nguyen, Blending is all you need: Data-centric ensemble synthetic data, Inform. Sci. 691 (2025) 121610.
[11] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.
[12] D. Elreedy, A.F. Atiya, F. Kamalov, A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning, Mach. Learn. (2023) 1–21.
[13] S. Rezvani, X. Wang, A broad review on class imbalance learning techniques, Appl. Soft Comput. 143 (2023) 110415.
[14] A. Gharizadeh, K. Abbasi, A. Ghareyazi, M.R. Mofrad, H.R. Rabiee, HGTDR: Advancing drug repurposing with heterogeneous graph transformers, Bioinformatics 40 (7) (2024).
[15] R.D. Camino, C.A. Hammerschmidt, et al., Oversampling tabular data with deep generative models: Is it worth the effort? in: J. Zosa Forde, F. Ruiz, M.F. Pradier, A. Schein (Eds.), Proceedings on "I Can'T Believe It's Not Better!" At NeurIPS Workshops, in: Proceedings of Machine Learning Research, Vol. 137, PMLR, 2020, pp. 148–157.

[3] https://scikit-learn.org/stable/auto_examples/classification/index.html.

[16] H. Ding, Y. Sun, Z. Wang, N. Huang, Z. Shen, X. Cui, RGAN-EL: A GAN and ensemble learning-based hybrid approach for imbalanced data classification, Inf. Process. Manage. 60 (2) (2023) 103235.
[17] T. Liu, J. Fan, G. Li, N. Tang, X. Du, Tabular data synthesis with generative adversarial networks: design space and optimizations, VLDB J. 33 (2) (2024) 255–280.
[18] A.X. Wang, S.S. Chukova, B.P. Nguyen, Synthetic minority oversampling using edited displacement-based k-nearest neighbors, Appl. Soft Comput. 148 (2023) 110895.
[19] A.X. Wang, S.S. Chukova, C.R. Simpson, B.P. Nguyen, Data-centric AI to improve early detection of mental illness, in: 2023 IEEE Statistical Signal Processing Workshop, SSP, IEEE, 2023, pp. 369–373.
[20] C. Vairetti, J.L. Assadi, S. Maldonado, Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification, Expert Syst. Appl. 246 (2024) 123149.
[21] A.X. Wang, S.S. Chukova, A. Sporle, B.J. Milne, C.R. Simpson, B.P. Nguyen, Enhancing public research on citizen data: An empirical investigation of data synthesis using Statistics New Zealand's Integrated Data Infrastructure, Inf. Process. Manage. 61 (1) (2024) 103558.
[22] H.M. Nguyen, E.W. Cooper, K. Kamei, Borderline over-sampling for imbalanced data classification, in: Proceedings of the Fifth International Workshop on Computational Intelligence & Applications, IEEE, 2009, pp. 24–29, http://dx.doi.org/10.1007/11538059_91.
[23] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1322–1328.
[24] A. Kulkarni, D. Chong, F.A. Batarseh, Foundations of data imbalance and solutions for a data democracy, in: Data Democracy, Elsevier, 2020, pp. 83–106, http://dx.doi.org/10.1016/B978-0-12-818366-3.00005-8.
[25] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 20–29, http://dx.doi.org/10.1145/1007730.1007735.
[26] J. Lucas, G. Tucker, R.B. Grosse, M. Norouzi, Don't blame the ELBO! a linear VAE perspective on posterior collapse, Adv. Neural Inf. Process. Syst. 32 (2019).
[27] N. Li, B. Guo, Y. Liu, Y. Ding, L. Yao, X. Fan, Z. Yu, Hierarchical Constrained Variational Autoencoder for interaction-sparse recommendations, Inf. Process. Manage. 61 (3) (2024) 103641.
[28] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, 2015, arXiv preprint arXiv:1511.05644.
[29] B. Paige, J.-W. Van De Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, P. Torr, et al., Learning disentangled representations with semi-supervised deep generative models, Adv. Neural Inf. Process. Syst. 30 (2017).
[30] S. Zhao, J. Song, S. Ermon, InfoVAE: Information maximizing variational autoencoders, 2017, arXiv preprint arXiv:1706.02262.
[31] J. Tomczak, M. Welling, VAE with a VampPrior, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2018, pp. 1214–1223.
[32] M.-f. Hu, Z.-y. Liu, J.-w. Liu, mcVAE: disentangling by mean constraint, Vis. Comput. 40 (2) (2024) 1229–1243.
[33] Y. Poels, V. Menkovski, VAE-CE: Visual contrastive explanation using disentangled VAEs, in: International Symposium on Intelligent Data Analysis, Springer, 2022, pp. 237–250.
[34] Z. Xie, C. Liu, Y. Zhang, H. Lu, D. Wang, Y. Ding, Adversarial and contrastive variational autoencoder for sequential recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 449–459.
[35] J. Aneja, A. Schwing, J. Kautz, A. Vahdat, A contrastive learning approach for training variational autoencoder priors, Adv. Neural Inf. Process. Syst. 34 (2021) 480–493.
[36] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional GAN, in: Advances in Neural Information Processing Systems, 2019, pp. 7335–7345.
[37] J.A. Green, Too many zeros and/or highly skewed? A tutorial on modelling health behaviour as count data with Poisson and negative binomial regression, Heal. Psychol. Behav. Med. 9 (1) (2021) 436–455.
[38] A. Asuncion, D. Newman, et al., UCI machine learning repository, 2007.
[39] L. Baccour, Amended fused TOPSIS-VIKOR for classification (ATOVIC) applied to some UCI data sets, Expert Syst. Appl. 99 (2018) 115–125.
[40] A.X. Wang, S.S. Chukova, B.P. Nguyen, Ensemble k-nearest neighbors based on centroid displacement, Inform. Sci. 629 (2023) 313–323.
[41] P. Razzaghi, K. Abbasi, J.B. Ghasemi, Multivariate pattern recognition by machine learning methods, in: Machine Learning and Pattern Recognition Methods in Chemistry from Multivariate and Data Driven Modeling, Elsevier, 2023, pp. 47–72.
[42] A.X. Wang, B.P. Nguyen, T. Elliott, J.F. Mbinta, A. Sporle, C.R. Simpson, Early detection of depression using machine learning and social well-being survey data, in: 2024 16th International Conference on Computer and Automation Engineering, ICCAE, IEEE, 2024, pp. 181–186.
[43] T. Saito, M. Rehmsmeier, The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets, PLoS One 10 (3) (2015) e0118432.